IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	Group Art Unit: 2446
	)	
Edward Eytchison *et al.*	)	Examiner: Ali, Farhad
	)	
Serial No. 10/763,866	)	**AMENDMENT AND RESPONSE TO**
	)	**OFFICE ACTION MAILED**
Filed: January 22, 2004	)	**June 22, 2009**
	)	
For: **METHODS AND**	)	162 North Wolfe Road
APPARATUSES FOR	)	Sunnyvale, California 94086
AUTOMATIC ADAPTATION OF	)	(408) 530-9700
DIFFERENT PROTOCOLS	)	
	)	Customer No. 28960

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

AMENDMENTS

Remarks/Arguments begin on page 2 of this paper.

REMARKS

Applicants respectfully request further examination and reconsideration in view of the arguments set forth fully below. Claims 1, 2, 4-11, 13, 15-20, and 22-24 were previously pending in this Application. Within the Office Action, Claims 1, 2, 4-11, 13, 15-20, and 22-24 have been rejected. Accordingly, Claims 1, 2, 4-11, 13, 15-20, and 22-24 are currently pending in the application.

Rejections Under 35 U.S.C. § 103

Within the Office Action, Claims 1, 2, 4-11, 13, 15-20, and 22-24 have been rejected under 35 U.S.C. §103(a) as being unpatentable over United States Patent Application Publication No. 2003/0204612 to Warren ("Warren") in view of United States Patent No. 5,623,695 to Lozinski ("Lozinski"). The Applicants respectfully disagree.

Warren

Warren teaches an abstraction device with a web services interface. The abstraction device receives web services commands in XML documents, translates the web services commands into one or more device commands in one or more device protocols, and transmits the device commands to one or more network elements. [Warren, ¶¶ 0006, 0024, 0025]. Warren teaches two different implementations of its abstraction device. [Fig. 1, elements 106a, 106b; Fig. 2, element 206]. In Fig. 1, "work manager 102 communicates using a web services protocol, and abstraction device 106 translates between the web services protocol and the protocols used by network elements 108." [Warren, ¶ 0020]. In this embodiment of Warren, the work manager is an application 102 which utilizes the web services protocol as an applications programming interface. Warren teaches that a network element 108 may be treated as a web service, and that a web service may be published, located, and invoked over a network. [Warren ¶ 0024]. The "manager [application] 102 may also store information about network elements." [Warren, ¶ 0023].

Warren's second embodiment of the abstraction device shows that web services interface 230 is the top-most layer of the abstraction device 206, further reinforcing that Warren teaches the known web services protocols XML and .NET as its application programming interfaces. [Warren, Fig. 2, elements 206, 230]. Thus, Warren does not teach a *single* application programming interface. In this embodiment, Warren also teaches a plurality of protocol convertors 238a-238e to convert commands 240 to device commands 248, and to convert device responses 268 to alerts 252. [Warren, Fig. 2]. Like the first embodiment, Warren again teaches a

plurality of protocol convertors simultaneously loaded into the abstraction device. Warren further teaches that database 236 may store information related to network elements 108. Such information includes network device IDs 254, network device types 256, command sequence information 258, control instruction information 260, script instruction information 262, field mapping information 264, and registration information. [Warren, ¶¶ 0053-0058]. Warren *does not teach searching* for anything. Warren does not teach *searching for at least one device based on content type*. Further, as discussed above, Warren teaches that its protocol translators are all always loaded. Warren does not teach *matching the detected protocol with a protocol translator*. Moreover, Warren teaches multiple application programming interface protocols and only a single application: the work manager 102. Warren does not teach a *single common application programming interface that is configured to be used by a plurality of applications.*

Lozinski

Lozinski teaches an application programming interface in a data processing system with multiple communication adapters for a particular service type, such as ISDN. [Lozinski, col. 1, lines 40-46]. Lozinski further teaches that "the complex configuration for the end user is avoided because no knowledge of the presence of one manufacturer's product by another is required." [Lozinski, col. 1, lines 65-57]. Thus, Lozinski teaches "translators" for multiple facilities through a common application programming interface (API), but the API is available only to a single application, 160. [Lozinski, Fig. 1, 120]. Lozinski further teaches that a lookup table is used to determine the entry point into the API corresponding to the desired adapter. [Lozinski, col. 1, lines 53-55]. The entry point is determined from information obtained during installation of the adapter. [Lozinski, col. 3, lines 34-45]. Therefore, Lozinski, like Warren, teaches that all translator modules are loaded into the API, and that only *known* devices are included within the scope of their teachings.

Within the Office Action, Lozinski is cited as teaching wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications. The Applicants respectfully disagree. Lozinski teaches *one* application accessing a plurality of similar devices using a common programming interface. Lozkinski does not teach a single programming interface that is configured to be used by a plurality of applications.

Combination of Warren and Lozinski

As discussed above, both Warren and Lozinski teach loading all of their protocol translators. Warren, Lozinski, and their combination, do not teach storing a list of network protocols available for use, detecting at least one device, detecting a protocol associated with the device, and matching the protocol with a protocol translator.

As discussed above, Warren teaches network elements which are known to the abstraction device by virtue of publishing their web services. Similarly, Lozinski teaches devices which are *known* to a common application interface by virtue of being "installed." Thus, Warren, Lozinski, and their combination do not teach *searching for at least one device based on a content type*.

As also discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications*.

In contrast to the teachings of Warren, Lozinski and their combination, the presently claimed invention teaches a lightweight, common application programming interface to be used by multiple applications in searching devices containing a type of content. [Present Specification, page 2, lines 14-15]. An application includes a list of available network protocols. In the presently claimed invention, at least one device is *searched for* based on a content type, the at least one device is detected, the protocol associated with the detected device is detected, the detected protocol associated with the device is matched with the device, a protocol translator module is matched with the protocol, *and the protocol translator is loaded*, then the translator module is used to translate a command formatted in the device protocol into a translated command formatted in a common application programming interface that is configured to be *used by a plurality of applications*. Thus, the presently claimed invention creates a lightweight run-time binding by *only loading* the protocol translator module(s) which correspond to the at least one device found by virtue of being *searched for* based on its content. [Present Specification, page 4, line 2].

Within the Office Action, Warren paragraph 0024 is cited as teaching "searching for at least one device based on a content type." The Applicants respectfully disagree. Warren does *not teach searching* for anything. As discussed above, in Warren, the location and type of web service (content) are *published* and are therefore already known. As also discussed above, Warren utilizes the information published by web services to populate its database of devices, device types, and other information. Nothing in Warren teaches detecting anything by *searching* for it. To the contrary, Warren is silent about network elements which do not publish web

- 4 -

services. Warren does not teach *searching for at least one device based on a content type.*

The independent Claim 1 is directed to a method comprising searching for at least one device based on a content type, detecting the at least one device, detecting a protocol associated with each device, matching the detected protocol with a protocol translator module, and using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface, wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications. As discussed above, Warren, Lozinski, and their combination do not teach *searching for at least one device based on a content type.* As also discussed above, Warren, Lozinski, and their combination, do not teach detecting at least one device, detecting a protocol associated with the device, and matching the protocol with a protocol translator. Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications.* For at least these reasons, the independent Claim 1 is allowable over the teachings of Warren, Lozinski and their combination.

Claims 2 and 4-6 are all dependent on the independent Claim 1. As described above, the independent Claim 1 is allowable over the teachings of Warren, Lozinski and their combination. Accordingly, Claims 2 and 4-6 are all also allowable as being dependent on an allowable base claim.

The independent Claim 7 is directed to a system comprising means for searching for at least one device based on a content type, means for detecting the at least one device, means for detecting a protocol associated with each device, means for matching the detected protocol with a protocol translator module, and means for using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface, wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications. As discussed above, Warren, Lozinski, and their combination do not teach *searching for at least one device based on a content type.* As also discussed above, Warren, Lozinski, and their combination, do not teach detecting a protocol associated with the device, and matching the protocol with a protocol translator. Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications.* For at least these reasons, the independent Claim 7 is allowable over the teachings of Warren, Lozinski and their combination.

The independent Claim 8 is directed to a method comprising searching for at least one service based on a content type, detecting at the least one service, detecting a protocol associated with each service, matching the detected protocol with a protocol translator module, and using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface, wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications. As discussed above, Warren, Lozinski, and their combination do not teach *searching for at least one device based on a content type*. As also discussed above, Warren, Lozinski, and their combination, do not teach detecting at least one device, detecting a protocol associated with the device and matching the protocol with a protocol translator. Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications*. For at least these reasons, the independent Claim 8 is allowable over the teachings of Warren, Lozinski and their combination.

The independent Claim 9 is directed to a method comprising searching for a specific device from a plurality of devices based on a content type, detecting the plurality of devices wherein each unique device communicates using a corresponding protocol, displaying an indication of each device if a protocol translator module is matched with the corresponding protocol, and translating a command formatted in the corresponding protocol into a translated command formatted in a common application programming interface through the protocol translator module, wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications. As discussed above, Warren, Lozinski, and their combination do not teach *searching for at least one device based on a content type*. As also discussed above, Warren, Lozinski, and their combination, do not teach detecting at least one device, detecting a protocol associated with the device and matching the protocol with a protocol translator. Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications*. For at least these reasons, the independent Claim 9 is allowable over the teachings of Warren, Lozinski and their combination.

Claims 10, 11, 13, 15 and 16 are all dependent on the independent Claim 9. As described above, the independent Claim 9 is allowable over the teachings of Warren, Lozinski and their combination. Accordingly, Claims 10, 11, 13, 15 and 16 are all also allowable as being dependent on an allowable base claim.

The independent Claim 17 is directed to a method comprising identifying a plurality of protocol translator modules wherein each protocol translator module is associated with a unique protocol, storing a list representing the plurality of protocol translator modules, displaying an indication of each device having a device protocol that is compatible with one of the plurality of protocol translator modules in the list, and translating a command formatted in the device protocol into a translated command formatted in a common application programming interface through one of the plurality of protocol translator modules, wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications. As discussed above, Warren, Lozinski, and their combination, do not teach storing a list of network protocols available for use and displaying an indication of each device having a device protocol that is compatible with one of the plurality of protocol translator modules. Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications*. For at least these reasons, the independent Claim 17 is allowable over the teachings of Warren, Lozinski and their combination.

Claims 18 and 19 are both dependent on the independent Claim 17. As described above, the independent Claim 17 is allowable over the teachings of Warren, Lozinski and their combination. Accordingly, Claims 18 and 19 are both also allowable as being dependent on an allowable base claim.

The independent Claim 20 is directed to a system comprising a plurality of applications configured for operating through a single, common application programming interface, a first device configured for operating using a first protocol, a second device configured for operating using a second protocol and a protocol translation layer configured for searching for a first protocol translation module corresponding to the first protocol and for searching for a second protocol translation module corresponding to the second protocol, the first protocol translation module and second protocol translation module stored in a list representing a plurality of protocol translator modules, wherein the protocol translation layer is configured to translate a first command formatted in the first protocol into a command formatted in the single, common application programming interface for use by one of the plurality of applications and to translate a second command formatted in the second protocol into a command formatted in the single, common application programming interface for use by another one of the plurality of applications. As discussed above, Warren, Lozinski, and their combination, do not teach storing a list of network protocols available for use and displaying an indication of each device having a device protocol that is compatible with one of the plurality of protocol translator modules.

Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications*. For at least these reasons, the independent Claim 20 is allowable over the teachings of Warren, Lozinski and their combination.

Claim 22 is dependent on the independent Claim 20. As described above, the independent Claim 20 is allowable over the teachings of Warren, Lozinski and their combination. Accordingly, Claim 22 is also allowable as being dependent on an allowable base claim.

The independent Claim 23 is directed to a network protocol translation system comprising a processor that executes a plurality of run time processes that use only a single application programming interface for network communication, wherein the processor enables at least one of the run time processes to communicate via a first network protocol by executing a first translation module that translates between the first network protocol and the single application programming interface and wherein the processor enables the at least one of the run time processes to communicate via a second network protocol, different from the first network protocol, and executing a second translation module that translates between the second network protocol and the application programming interface, further wherein the first translation module and second translation module are stored in a list representing a plurality of protocol translator modules. As discussed above, Warren, Lozinski, and their combination, do not teach storing a list of network protocols available for use and displaying an indication of each device having a device protocol that is compatible with one of the plurality of protocol translator modules. Further, as discussed above, Warren, Lozinski, and their combination, do not teach *a single application programming interface that is configured to be used by a plurality of applications*. For at least these reasons, the independent Claim 23 is allowable over the teachings of Warren, Lozinski and their combination.

The independent claim 24 is directed to a method, executed on a computing platform, comprising the acts of executing a plurality of run time processes that uses only a single application programming interface for network communication, enabling at least one of the run time processes to communicate via a first network protocol by executing a first translation module that translates between the first network protocol and the single application programming interface and enabling the at least one of the run time processes to communicate via a second network protocol, different from the first network protocol, by executing a second translation module that translates between the second network protocol and the single application programming interface, wherein the first translation module and second translation module are stored in a list representing a plurality of protocol translator modules. As discussed above,

Warren, Lozinski, and their combination do not teach executing a plurality of run time processes that uses only a single application programming interface for network communication. For at least these reasons, the independent Claim 24 is allowable over the teachings of Warren, Lozinski and their combination.

For the reasons given above, the Applicant respectfully submits that pending Claims are all in condition for allowance, and allowance at an early date would be appreciated. Should the Examiner have any questions or comments, he is encouraged to call the undersigned at (408) 530-9700 to discuss them so that any outstanding issues can be expeditiously resolved.

Respectfully submitted,

HAVERSTOCK & OWENS LLP

Dated: September 18, 2009   By: /Jonathan O. Owens/

                                            Jonathan O. Owens

                                            Reg. No. 37,902

                                            Attorneys for Applicant(s)